Teaching statistical physics by thinking about models and algorithms

Jan Tobochnik*

Department of Physics, Kalamazoo College, Kalamazoo, Michigan 49006

Harvey Gould†

Department of Physics, Clark University, Worcester, Massachusetts 01610

Abstract

We discuss several ways of illustrating fundamental concepts in statistical and thermal physics by considering various models and algorithms. We emphasize the importance of replacing students' incomplete mental images by models that are physically accurate. In some cases it is sufficient to discuss the results of an algorithm or the behavior of a model rather than having students write a program.

I. INTRODUCTION

Mathematics is both the language of physics and a calculational tool. For example, the statements $\mathbf{a} = \mathbf{F}/m$ and $\nabla \cdot \mathbf{B} = 0$ express the ideas that acceleration is the result of forces and magnetic field lines exist only as closed loops. The fundamental thermodynamic relation $dS = (1/T)dU + (P/T)dV - (\mu/T)dN$ implies that there exists an entropy function that depends on the internal energy U, the volume V, and the number of particles N. It also tells us that the temperature T, the pressure P, and the chemical potential μ determine how the entropy changes.

Although textbooks and lectures describe the meaning of mathematical relations in physics, students frequently do not understand their meaning because there are few related activities that undergraduate students can do. Instead students frequently use mathematics as a calculational tool for problems which lead to little physical understanding. To introductory students $\mathbf{a} = \mathbf{F}/m$ is just one of many algebraic relations which needs to be manipulated. For more advanced students it is a differential equation which needs to be solved.

The availability of symbolic manipulation software and calculators that can do much of the mathematical manipulations which students have been traditionally asked to do means that instructors need to think carefully about what are the appropriate activities for students. How many of the problems at the back of textbook chapters teach useful skills or help students learn physics? Which skills are important? How much, if any, understanding is lost if more of the calculations are done with the aid of a computer? Is most of the physics in the setup of the problem and in the analysis of the results? Should we spend more time teaching students to use software packages more effectively?

In addition to challenging the way we teach the traditional physics curriculum, there is a growing interest in including more computation into the curriculum. Many physics teachers view computation as another tool analogous to mathematical tools such as those used to solve algebraic and differential equations. The theory is the same and computation is needed only when exact solutions are not available. Because the latter provide useful illustrations of the theory, computation need be only a minor part of the undergraduate (and even graduate) curriculum. This situation is the common practice at most institutions.

A popular rationale for more computation in the curriculum is that it allows us to do

more realistic problems, is important in scientific research, and provides a useful tool for later employment. These reasons are all valid, but they might be less important than they seem. Although computation can allow us to consider more realistic problems, the consideration of such problems usually requires a much greater understanding of the system of interest than most undergraduates have the background and time to learn. Knowing a programming language is useful in employment, but might become less important as more and more work is done by higher level languages targeted toward specific applications. And even though computation is ubiquitous in research, more of it is being done using software packages. Just as few experimentalists need to know the details of how electronic instrumentation is constructed, few scientists need to know the details of how software packages are written. Moreover there usually is not enough time for students to write their own programs except in specialized courses in computational physics and simulation.

In this paper we argue that computation should be incorporated into the curriculum because it can elucidate the physics. As for mathematics, computation is both a language and a tool. In analogy to models expressed in mathematical statements there are models expressed as algorithms. In many cases the algorithms are explicit implementations of the mathematics. For example, writing a differential equation is almost the same as writing a sequence of rate equations for the variables in a computer program. An advantage of the computational approach is that it is necessary to be explicit about which symbols represent variables and which represent initial conditions and parameters.

In other cases the algorithm does not look like a traditional mathematical statement. For example, the Monte Carlo algorithms used in statistical physics do not look like expressions for the partition function or the free energy. There also are models such as cellular automata that have no counterpart in traditional mathematics.

In the following we will discuss some examples of how the consideration of algorithms and models can help students and instructors understand some fundamental concepts in physics. Because computation has had a great influence on statistical physics, we will focus our attention on this area. It is also the area in which we have the most expertise.

II.   APPROACH TO EQUILIBRIUM

A basic understanding of probability is necessary for understanding the statistical behavior underlying thermal systems. Consider the following question, which can be stated simply in terms of an algorithm.[1] Imagine two bags and a large number of balls. All the balls are initially placed in bag one. Choose a ball at random and move it to the other bag. After many moves, how many balls are in each bag? Many students will say that the first bag will have more balls on the average.[2] In this case it might be useful to show a simulation of the ball swapping process.[3] It is then a good idea to ask students to sketch the number of balls in one bag as a function of time before showing the simulation. We can then discuss why the mean number of balls in each bag eventually becomes time independent. In particular, we can illustrate the presence of fluctuations during relaxation and in equilibrium and how the results depend on the number of particles. We can illustrate many of the basic features of thermal systems including the concepts of microstate and macrostate, the history independence of equilibrium (the equilibrium macrostate is independent of the initial conditions), ensemble and time averages, and the need for a statistical approach. We can also discuss why fluctuations in macroscopic systems will be negligible for most thermal quantities.

The approach to equilibrium can be repeated with a molecular dynamics simulation (see Sec. III) in which particles are initially confined to one part of a container. After the release of an internal constraint the particles eventually are uniformly distributed on the average throughout the container. It is important in this example to show the students the basic algorithm and some simple computer code, so that they are convinced that there is no explicit "force" pushing the system toward equilibrium, but rather that equilibrium is a result of random events. The idea is to make explicit that the general behavior of systems with many particles is independent of the details of the inter-particle interactions.

III.   MOLECULAR DYNAMICS

If students are asked what happens to the temperature when a gas is compressed, they will likely say that it increases. Their microscopic explanation will likely be that the molecules rub against each other and "give off heat."[2,4] These students have a mental model which is similar to a system of marbles with inelastic collisions.

This granular matter model of a gas and liquid is appropriate for the types of non-thermal macroscopic systems that students experience in everyday life such as cereals, rice, and sand. For molecular systems the collisions are elastic, and a hard sphere model[5] with elastic collisions is useful for understanding the properties of fluids and solids. Thus, student intuition has some value, but is limited in its ability to account for much of the phenomena of thermal systems. In particular, the only relevance of the temperature in hard sphere models is to set the the natural energy scale.

A concrete and realistic model of thermal systems is provided by thinking about molecular dynamics with continuous inter-particle potentials for a system with a fixed number of particles and fixed volume.[6] Each particle is subject to the force of every other particle in the system, with nearby particles usually providing most of the force. The dynamics of the ith particle of mass m_i is determined by Newton's second law, $\mathbf{a}_i = \mathbf{F}_{\mathrm{net},i}/m_i$, which can be integrated numerically to obtain the position and velocity of each particle.[8] Students should be asked to think about the appropriate inter-molecular force law and what would happen in the simulation and a real thermal system. For example, what is different about a gas and a liquid? This question will lead students to conclude that there must be an attractive contribution to the force law. Also systems do not completely collapse and thus there must be a repulsive part as well. Students can then be led to conclude that the force law must look something like the force law derivable from the Lennard-Jones potential.

Next ask students about pressure, and you will likely not receive a coherent answer. Even if students know that pressure is force per unit area, it is likely that they will not be able to explain what that means in terms of a microscopic model of a gas. The simplest way to determine the pressure is to compute the momentum transfer across a surface. Without doing the simulation we can see that because momentum is proportional to velocity, faster particles will lead to greater momentum transfer per unit time. Then we can discuss what other macroscopic quantities increase with particle speed. Students will typically bring up temperature because they associate temperature with kinetic energy. From this discussion they can conclude that the pressure and temperature depend on the mean speed of the particles. Also, if the density increases at a given temperature, more particles will cross a given surface, and hence the pressure increases with the density. Students can then conclude that there are two independent quantities, the density and the temperature, which determine the pressure.

Because the total energy is conserved in an isolated container, students can understand that the total energy consists of both potential and kinetic energy. At this point some discussion is needed to help students understand that there is potential energy that has nothing to do with gravity. Once that understanding is reached, it is easy to understand that the kinetic energy does not remain constant, and thus there will be small fluctuations in the temperature. Similar reasoning can lead to the idea of pressure fluctuations. Thus, by imagining the particles in a molecular dynamics simulation the concept of thermal fluctuations can be inferred. This result can be related to the fluctuations discussed in the ball swapping model in Sec. II.

What is the role of temperature? We will discuss temperature again in Sec. IV, but here we consider how molecular dynamics can be used to think about its meaning. Imagine two solids such that initially the particles in one solid are moving much faster than the particles in the other solid. The two solids are placed in thermal contact so that the particles interact with each other across the boundary between the two solids. This model provides a concrete realization of thermal contact. Particles at the boundary will exchange momentum and energy with each other. The faster particles will give energy to the slower ones and energy will be transferred from the solid with the faster particles to the one with the slower particles. The net energy transfer will cease when all the particles have the same mean kinetic energy per particle, but not the same total energy per particle, potential energy per particle or mean speed. Thus by thinking about this system we can gain insight about the connection between kinetic energy and temperature. More importantly, we can emphasize that temperature is the quantity that becomes the same when two systems are placed in thermal contact.

Our experience has been that a consideration of molecular dynamics in various contexts over several weeks leads to most students replacing their granular matter model by one in which energy is conserved.

IV. AN IDEAL THERMOMETER

In a previous paper[9] we discussed the *demon algorithm*, which can be used as a model of an ideal thermometer and as a chemical potential meter. In a computer simulation the demon is an extra degree of freedom which can exchange energy or particles with a system. Energy is exchanged by making a random change in one or more degrees of freedom of the

system. If the change increases the energy, the change is accepted only if the demon has enough energy to give the system. If the trial move decreases the energy of the system, the extra energy is given to the demon. In a similar way the demon can exchange particles with the system. The energy distribution of the demon for a given number of particles in the system is the Boltzmann distribution from which the temperature can be extracted because the demon and system are in thermal equilibrium. If particle and energy exchanges are allowed, the distribution of energy and particles held by the demon is a Gibbs distribution from which the chemical potential can be extracted.

A discussion of the demon is a good way of introducing the concepts of system, heat bath, and ensembles. The demon algorithm simulates the microcanonical ensemble because the total energy of the system plus the demon is fixed, and the demon is a negligible perturbation of the system. An alternative interpretation is that the demon is the system of interest and the remaining particles play the role of the heat bath as in a canonical ensemble. The demon is unusual because its energy is both the energy of the system and the energy of a microstate. Similarly, the state of the demon is both a microstate and a macrostate. In realistic thermal systems there are many microstates corresponding to a given macrostate, and thus there are distinctions between a macrostate, a microstate, and a single particle state. The demon algorithm provides a concrete example for discussing such concepts.

V.   MARKOV CHAINS AND THE METROPOLIS ALGORITHM

One of the most common algorithms used to simulate thermal systems is the Metropolis algorithm.[10] In this section we discuss the underlying theory behind this algorithm, which will allow us to introduce concepts such as probability distributions, the Boltzmann distribution, Markov chains, the partition function, sampling, and detailed balance. In Sections VI-VIII we will extend these ideas to newer algorithms that are currently of much interest in research.

Most students are familiar with probabilities in the context of dice and other simple games of chance, but they have difficulty understanding what probabilities mean in the context of thermal systems.

Students are familiar with systems that evolve in time according to deterministic equations such as Newton's second law. We can also consider stochastic processes such as those

used in Monte Carlo simulations (see Sec. II). In this case we consider an ensemble of identical copies of a system. These copies have the same macroscopic parameters, interactions, and constraints, but the microscopic states (such as the positions and momenta of the particles) are different in general. We imagine a probabilistic process which changes the microscopic states of the members of the ensemble. An example is a Markov process for which the probability distribution of the states of the ensemble at time t depends only on the probability distribution of the ensemble at the previous time; that is, the system has no long term memory. A Markov process can be represented by the equation:

$$P(t + \Delta t) = \pi P(t), \tag{1}$$

where $P(t)$ is a column matrix of n entries, each one representing the probability of a microstate; π is a matrix of transition probabilities. The ith entry of $P(t)$ gives the fraction of the members of the ensemble that are in the ith state at time t. By repeatedly acting on P with the transition matrix we arrive at the distribution at any later time t. If a system is in a stationary state, then

$$P(t + \Delta t) = P(t). \tag{2}$$

How can we be sure that the Markov process we use in our simulations will lead to the theoretically correct equilibrium distribution? We begin with the detailed balance condition

$$\pi_{i \to j} P_i = \pi_{j \to i} P_j, \tag{3}$$

where $\pi_{i \to j}$ is the probability of the system going from state i to state j, and P_i is the probability that the system is in state i. Detailed balance puts a constraint on the transition probabilities. As we will show, if the stochastic process satisfies detailed balance, then a system which is in a stationary state will remain in that state.

We now show that this property is satisfied for an equilibrium system at temperature $T = 1/k\beta$ for which the probability of a microstate is given by the Boltzmann distribution. If we substitute the Boltzmann distribution $P_i = e^{-\beta E_i}/Z$ into Eq. (3), we obtain

$$\pi_{i \to j} \, e^{-\beta E_i} = \pi_{j \to i} \, e^{-\beta E_j}, \tag{4}$$

where E_i is the energy of state i and k is Boltzmann's constant. Note that the partition function $Z = \sum_i e^{-\beta E_i}$ does not appear in Eq. (4). This absence is important because Z is usually not known.

8

We use Eq. (4) and the Boltzmann distribution for the desired stationary state to calculate the right-hand side of each row of Eq. (1):

$$P_i(t + \Delta t) = \sum_j \pi_{j \to i} P_j(t). \tag{5}$$

We next use the detailed balance condition in Eq. (4) to obtain

$$P_i(t + \Delta t) = \sum_j \left[\pi_{i \to j} e^{-\beta(E_i - E_j)} \right] P_j(t). \tag{6}$$

We want to see if an ensemble that is described by the Boltzmann distribution will remain in this distribution. We replace P_j in Eq. (6) by the Boltzmann distribution $e^{-\beta E_j}/Z$ so that

$$P_i(t + \Delta t) = \sum_j \left[\pi_{i \to j} e^{-\beta(E_i - E_j)} \right] \frac{e^{-\beta E_j}}{Z}. \tag{7}$$

We can simplify Eq. (7) by writing

$$P_i(t + \Delta t) = \frac{e^{-\beta E_i}}{Z} \sum_j \pi_{i \to j}, \tag{8}$$

where we have taken the factor that does not depend on j out of the sum. Because the system has probability unity of going from the state i to all possible states j, the sum over j must equal unity, and thus Eq. (8) becomes

$$P_i(t + \Delta t) = \frac{e^{-\beta E_i}}{Z}. \tag{9}$$

Thus, we have shown that if we start with an ensemble distributed according to the Boltzmann distribution and use a transition probability that satisfies detailed balance, then the resulting ensemble remains in the Boltzmann distribution. Note that detailed balance is sufficient, but we have not shown that detailed balance is necessary for an ensemble to remain in a stationary state, and it turns out that it is not.

Although detailed balance specifies the ratio $\pi_{i \to j}/\pi_{j \to i}$, it does not specify $\pi_{i \to j}$ itself. There is much freedom in choosing $\pi_{i \to j}$, and the optimum choice depends on the nature of the system being simulated. The earliest and still a popular choice is $\pi_{i \to j} = A_{i \to j} W_{ij}$, where W_{ij} is the probability of making an arbitrary trial move and $A_{i \to j}$ is the Metropolis acceptance probability given by

$$A_{i \to j} = \min \left\{ 1, e^{-\beta(E_j - E_i)} \right\}. \tag{10}$$

Equation (10) can be shown to satisfy detailed balance if $W_{ij} = W_{ji}$ (see Problem 1b). For example, for an Ising model with N spins choosing a spin at random implies $W_{ij} = W_{ji} = 1/N$.

Our discussion of probability has been a mixture of some abstract ideas from probability theory and the example of the Metropolis algorithm. The discussion can be made more concrete by considering a specific model, such as the Ising model of magnetism or a Lennard-Jones system of particles.[11]

VI. DIRECT ESTIMATION OF THE DENSITY OF STATES

The density of states $g(E)$ is defined[12] so that $g(E)\Delta E$ is the number of microstates with energy between E and $E + \Delta E$. For most students the density of states is an abstract quantity and many confuse the density of states of a many body system with the single particle density of states. The following discussion provides an algorithm which makes the meaning of $g(E)$ more concrete.

If the density of states $g(E)$ is known, we can calculate the mean energy and other thermodynamic quantities at any temperature from the relation

$$<E> = \sum_E E\, P(E), \tag{11}$$

where the probability $P(E)$ that a system in equilibrium with a heat bath at temperature T has energy E is given by

$$P(E) = \frac{g(E)\, e^{-\beta E}}{Z}, \tag{12}$$

and $Z = \sum_E g(E)\, e^{-\beta E}$. Hence, the density of states is of much interest.

Extracting the density of states from a molecular dynamics simulation or from the Metropolis and similar algorithms is very difficult. For example, suppose that we try to determine $g(E)$ by doing a random walk in energy space by flipping the spins in the Ising model at random and accepting all microstates that are obtained. The histogram of the energy, $H(E)$, the number of visits to each possible energy E of the system, would converge to $g(E)$ if the walk visited all possible configurations. In practice, it would be impossible to realize such a long random walk given the extremely large number of possible configurations in even a small system.

Recently a Monte Carlo algorithm due to Wang and Landau[13] has been developed for determining $g(E)$ directly. The idea is to simulate a system by making changes at random, but to sample energies that are more probable less often so that $H(E)$ becomes approximately independent of E. The acceptance criteria is given by

$$A_{i \to j} = \min \left\{ 1, \frac{\tilde{g}(E_i)}{\tilde{g}(E_j)} \right\}. \tag{13}$$

where $\tilde{g}(E)$ is the running estimate of $g(E)$. This acceptance criteria favors energies which have a low density of states, and because these states are visited more often, the result is that a histogram of the energy of the visited configurations will be approximately independent of E or "flat." It is possible to show (see Problem 1c) that the probability of the ith microstate is proportional to $1/g(E_i)$ in the stationary state.

How do we implement Eq. (13) when we don't know $g(E)$, which is the goal of the simulation? The second part of the algorithm is to make an initial guess for $\tilde{g}(E)$ and then improve the estimate of $\tilde{g}(E)$ as the simulation proceeds. The simplest guess is to let $\tilde{g}(E) = 1$ for all E and then to update $\tilde{g}(E)$ and $H(E)$ after each trial move:

$$\tilde{g}_{t+1}(E) = f \tilde{g}_t(E), \tag{14a}$$

or

$$\ln \tilde{g}_{t+1}(E) = \ln \tilde{g}_t(E) + \ln f, \tag{14b}$$

with

$$H_{t+1}(E) = H_t(E) + 1, \tag{15}$$

and $f > 1$; t represents the number of updates. (The value of E in Eqs. (14) and (15) is unchanged if the trial move is not accepted.) Because $\tilde{g}(E)$ increases rapidly with E, we need to use $\ln \tilde{g}(E)$ to implement this algorithm on a computer.

The combination of Eqs. (13) and (14) forces the system to visit less explored energy regions due to the bias in the acceptance probability in Eq. (13). For example, if the current estimate $\tilde{g}(E)$ is too low, moves to states with a lower value of $\tilde{g}(E)$ have a greater probability. In this way the values of $\tilde{g}(E)$ will gradually approach the "true" $g(E)$.

Initially we choose f to be large (typically $f = e \approx 2.7$) so that $g(E)$ will change rapidly. After many moves the histogram $H(E)$ will become approximately flat, and we then decrease f so that $\tilde{g}(E)$ doesn't change as much. The usual procedure is to let $f \to \sqrt{f}$ and continue the updates of $\tilde{g}(E)$ and the changes in f until $f - 1 \sim O(10^{-8})$.

As can be seen in Eq. (11) the only quantity that differentiates the mean energy and the specific heat of one system from another is the density of states. However, the density of states of many systems is qualitatively similar. This similarity is illustrated in Fig. 1, which shows the density of states for 64 Ising spins in one, two, and three dimensions. More insight can be gained from the plot in Fig. 2 of the probability $P(E)$ superimposed on a plot of $\ln g(E)$. The plot shows the range of values of E that are important for a given temperature. The competition between the increase of $g(E)$ with E and the decrease of the Boltzmann factor leads to the peak in $P(E)$.

One of the interesting features of the Wang-Landau algorithm is that it samples states that are of little interest for thermal systems, unlike the Metropolis algorithm which rarely samples such states. For example, an application of the Wang-Landau algorithm to the Ising model would lead to a parabolic-like curve for $\ln g(E)$ with a maximum at $E = 0$ as shown in Fig. 1. Positive energy states, with most of the nearest neighbor spins of opposite sign, are present. Because large positive energy states would not be observed in a Metropolis simulation of the Ising model, it is clear that we are measuring a temperature independent property that depends only on the nature of the system.

VII. DIRECT ESTIMATION OF T(E) IN THE MICROCANONICAL ENSEMBLE

Another way to determine $g(E)$ is to exploit the correspondence between the density of states and the thermodynamic temperature $T(E)$ and to update the latter rather than $g(E)$.[14] The approach is based on the relation between the microcanonical entropy, $S(E) = \ln g(E)$ (with Boltzmann's constant $k = 1$), and the inverse temperature, $\beta(E) = 1/T(E) = \partial S/\partial E$. The main virtue of this method of determining $g(E)$ in the present context is that it is an application of the relation between the entropy and the temperature.

For simplicity, we discuss the algorithm for the Ising model for which the values of the energy are discrete. We label the possible energies of the system by the index j with $j = 1$ corresponding to the ground state and write the temperature estimate $\tilde{T}(E_j)$ as $\tilde{T}_j$. After a trial move to state j we update the entropy S_j as for the Wang-Landau algorithm and also update the temperature estimate at $j \pm 1$. From Eq. (14a) or

$$\tilde{S}_{j,t+1} = \tilde{S}_{j,t} + \ln f, \tag{16}$$

we can write the central difference approximation for the inverse temperature β as

$$\frac{\partial \tilde{S}}{\partial E}\bigg|_{E=E_{j,t+1}} = \tilde{\beta}_{j,t+1} \approx [\tilde{S}_{j+1,t} - \tilde{S}_{j-1,t}]/(2\Delta E), \tag{17}$$

where ΔE is the energy difference between state j and states $j \pm 1$. (For the Ising model on a square lattice, $\Delta E = 4J$. In general, we would choose a larger bin size so that several states would correspond to one bin.) The energy E_j in Eq. (17) is the value of the energy of the system after a trial move. On a visit to E_j we use the updated value of $\tilde{S}_j$ and the unchanged values of $\tilde{S}_{j\pm2}$ to determine new estimates for the temperature. From Eq. (16) we have $\tilde{\beta}_{j+1,t+1} = [\tilde{S}_{j+2,t} - \tilde{S}_{j,t+1}]/(2\Delta E) = \tilde{\beta}_{j+1} - \delta f$, and $\tilde{\beta}_{j-1,t+1} = [\tilde{S}_{j,t+1} - \tilde{S}_{j-2,t}]/(2\Delta E) = \tilde{\beta}_{j-1,t} + \delta f$ with $\delta f = \ln f/2\Delta E$. Hence, we obtain

$$\tilde{T}_{j\pm1,t+1} = 1/\beta_{j\pm1} = \alpha_{j\pm1,t}\tilde{T}_{j\pm1,t}, \tag{18}$$

where $\alpha_{j\pm1,t} = 1/[1 \mp \delta f \tilde{T}_{j\pm1,t}]$. Note that $\tilde{T}_{j-1,t+1}$ is decreased and $\tilde{T}_{j+1,t-1}$ is increased while $\tilde{T}_j$ is unchanged. In this way $\tilde{T}(E)$ will converge to a monotonically increasing function of the energy. It is best to restrict the updates to a finite range of temperatures between $T_{\min}$ and $T_{\max}$.

The acceptance probability of the trial moves is given by Eq. (13) so we also need to update the entropy as is done for the Wang-Landau algorithm. The values of f are changed as for the Wang-Landau algorithm. An example of the converged $\tilde{T}(E)$ for the Ising model is given in Fig. 3.

Once $\tilde{T}(E)$ converges, the entropy estimate is found by integrating $\tilde{T}(E)$ with respect to E. In the simplest interpolation method the entropy is given by

$$S(E) = \sum_{j=j_{\min}}^{j} \tilde{\beta}_j \Delta E. \tag{19}$$

These values of $S(E)$ are then used to obtain $g(E)$, and the thermodynamic properties of the system are determined from Eq. (11). Because the continuum entropy $S(E)$ can be obtained by integrating the interpolated T_j, the updating of $T(E)$ as in Eq. (18) is especially useful for systems where the energy changes continuously. Even more interesting is the use of Eq. (18) with molecular dynamics simulations at various temperatures.[14,15]

VIII. DIRECT MEASUREMENT OF THE PARTITION FUNCTION

Another Monte Carlo method combines the Metropolis algorithm with the Wang-Landau method to directly compute the partition function Z at all temperatures of interest.[16] The method uses two types of trial moves: a standard Metropolis Monte Carlo move such as a flip of a spin which changes the energy at fixed temperature, and a move to change the temperature at fixed energy. The acceptance rule is given by

$$A_{i \to j} = \min \left\{ 1, \frac{e^{\beta_i E_i} / \tilde{Z}(\beta_i)}{e^{\beta_j E_j} / \tilde{Z}(\beta_j)} \right\}, \qquad (20)$$

where E_i is the energy of the current configuration, β_i is the current inverse temperature and $\tilde{Z}(\beta_i)$ is the current estimate of the partition function for inverse temperature β_i. For an energy move, $\beta_i = \beta_j$, and for a temperature move $E_i = E_j$. At each trial move a decision to choose a temperature instead of an energy move is made with a fixed probability such as 0.1%. To update the partition function we use the same procedure as for the density of states in the Wang-Landau method:

$$\ln \tilde{Z}(\beta) \to \ln \tilde{Z}(\beta) + \ln f. \qquad (21)$$

This procedure will lead to a stationary state with probability

$$P_i = e^{\beta_i E_i} / Z(\beta_i). \qquad (22)$$

A plot of $\ln[Z(T)/Z(T = 0.1]/N$ for the two-dimensional Ising model with $N = 32 \times 32$ spins is shown in Fig. 4. We rarely show plots of partition functions in a thermal physics course. The plot shows that $\ln Z$ does not change very much at low temperatures, increases rapidly near the phase transition, and then increases slowly for higher temperatures.

IX. SUMMARY

Our focus has been on understanding important concepts in statistical mechanics by considering simulations of concrete models of thermal and statistical systems. Instructors can choose how to use simulations in their courses. For example, students can be asked to write programs with the use of templates.[17] Another strategy is to have students run existing programs and modify some of the parameters and explain the results.[3]

Computers are omnipresent in physics research. Much of this use is for data analysis, symbolic algebra (for example, to calculate Feynman diagrams), and the control of experiments. Statistical physics is an area where the development of new algorithms and simulations has qualitatively changed the kinds of systems and problems that can be considered. These developments make it even more important to think about the ways that computers should change the way we teach thermal and statistical physics.

X. SUGGESTIONS FOR FURTHER STUDY

1. Detailed balance.

   (a) We showed that the detailed balance condition in Eq. (4) ensures that the Boltzmann probability distribution is stationary. Show that the detailed balance condition, Eq. (3), ensures that the distribution P_i is stationary.

   (b) Show that the Metropolis algorithm satisfies detailed balance if $W_{ij} = W_{ji}$. Show that the symmetric acceptance probability,

   $$A_{i \to j} = \frac{e^{-\beta E_j}}{e^{-\beta E_j} + e^{-\beta E_i}}, \qquad (23)$$

   also satisfies detailed balance.

   (c) Show that the stationary probability distribution for the Wang-Landau algorithm, Eq. (13), is $P_i = c/g(E_i)$, where c is a constant independent of E.

2. Approach to equilibrium.

   (a) Write a program that implements the balls and bags example discussed in Sec. II or use the application/applet available from $\langle$stp.clarku.edu$\rangle$.[3]

   (b) Plot the number of particles on the left side of the container as a function of time. How long does it take for a system of $N = 64$ balls to come to equilibrium? Then estimate this time for $N = 128$, 256, and 512 balls.

   (c) Once the system reaches equilibrium there are still fluctuations. Find a relation for the magnitude of the fluctuations as a function of N.

(d) Does there exist an initial configuration of the balls that comes to a different equilibrium state than the ones you have simulated so far? Why is the answer to this question important in statistical mechanics?

3. Wang-Landau algorithm

(a) Calculate by hand the density of states for $N = 6$ Ising spins in one dimension with toroidal boundary conditions. The minimum energy is -6 and the maximum energy is $+6$. There are $2^6 = 64$ microstates.

(b) Write a program that uses the Wang-Landau algorithm to determine the density of states for the one-dimensional Ising chain and compare your results with your hand calculation.

4. Additional problems related to the topics in this article as well as many other topics in statistical physics can be found at ⟨stp.clarku.edu⟩ or EPAPS.[3]

Acknowledgments

---

[*] Electronic address: jant@kzoo.edu

[†] Electronic address: hgould@clarku.edu

[1] See for example, R. M. Eisberg, *Applied Mathematical Physics with Programmable Pocket Calculators* (McGraw-Hill, New York, 1976), Chap. 7, and R. M. Eisberg and L. S. Lerner, *Physics: Foundations and Applications* (McGrawHill, New York, 1981), Sec. 18-7.

[2] These observations are based on interviews by Jan Tobochnik with students who have taken standard introductory physics and chemistry courses.

[3] Approximately forty (open source) simulations of systems of interest in statistical and thermal physics are available from ⟨stp.clarku.edu⟩. The simulations come with related instructional

material that can be easily modified by instructors. Also available are notes by the authors on statistical and thermal physics which incorporate simulations and discussions of algorithms throughout the text. You can also obtain these simulations from EPAPS Document No. ***. This document can be reached through a direct link in the online articles HTML reference section or via the EPAPS homepage (http://www.aip.org/pubservs/epaps.html).

[4] M. E. Loverude, C. H. Kautz, and P. R. L. Heron, "Student understanding of the first law of thermodynamics: Relating work to the adiabatic compression of an ideal gas," Am. J. Phys. **70** (2), 137–148 (2002).

[5] Hard spheres were first simulated using molecular dynamics by B. J. Alder and T. E. Wainwright, "Phase transitions for a hard sphere system," J. Chem. Phys. **27**, 1208–1209 (1957).

[6] There are many books and articles on molecular dynamics. See for example, D. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 2004), 2nd ed., or Ref. 7, Chap. 8.

[7] H. Gould, J. Tobochnik, and W. Christian, *Introduction to Computer Simulation Methods, Applications to Physical Systems* (Addison-Wesley, San Francisco, 2007).

[8] Although the simple Euler algorithm is frequently used in introductory courses to numerically integrate Newton's equations of motion, it is not adequate for molecular dynamics. The most commonly used algorithm, the Verlet algorithm,[6,7] is easy to understand and implement.

[9] J. Tobochnik, H. Gould, and J. Machta, "Understanding the temperature and the chemical potential using computer simulations," Am. J. Phys. **73** (8), 708–716 (2005).

[10] There are many excellent books on the Metropolis algorithm and Monte Carlo methods in statistical physics. See for example, M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, Oxford, 1999) and D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, Cambridge, 2005), 2nd ed.

[11] See for example, Ref. 7, Chap. 15 for a discussion of Monte Carlo simulations of the Ising model and Lennard-Jones systems.

[12] In this case as for the Ising model the possible values of E are discrete and hence $g(E)$ is the number of states with energy E. However, $g(E)$ is commonly referred to as the density of states independently of whether the energy is a continuous or discrete variable or if the possible energy values are binned.

[13] D. P. Landau, Shan-Ho Tsai, and M. Exler, "A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling," Am. J. Phys. **72**, 1294–1302 (2004).

[14] J. Kim, J. E. Straub, and T. Keyes, "Statistical temperature Monte Carlo and molecular dynamics algorithms," Phys. Rev. Lett. **97**, 050601-1–4 (2006).

[15] J. Kim, J. E. Straub, and T. Keyes, "Statistical temperature molecular dynamics: Application to coarse-grained β-barrel-forming protein models," J. Chem. Phys. **126**, 135101-1–4 (2007).

[16] Cheng Zhang and Jianpeng Ma, "Simulation via direct computation of partition functions," Phys. Rev. E **76**, 036708-1–5 (2007).

[17] Easy Java simulations (Ejs) and VPython are easy to use by students with little knowledge of programming. See ⟨`http://www.um.es/fem/Ejs/`⟩ and ⟨`http://vpython.org/`⟩.

[18] R. B. Pearson, "Partition function of the Ising model on the periodic $4 \times 4 \times 4$ lattice," Phys. Rev. B **26**, 6285–6290 (1982).
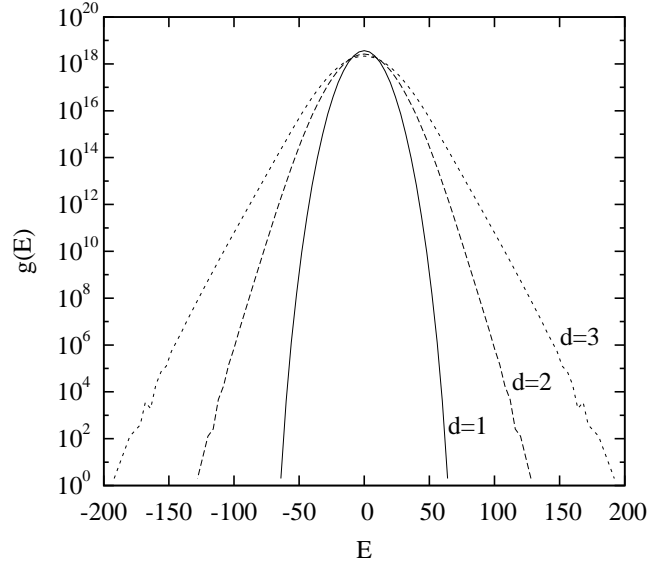
Figures

FIG. 1: Semi-log plot of the exact densities of states for $N = 64$ spins for one, two, and three dimensions. The results for three dimensions were generated using the method discussed in Ref. 18. The results of the Wang-Landau algorithm are indistinguishable from the exact results for these small systems. Note the tiny deviations from a smooth curve at $|E| \approx 120$ for $d = 2$ and $|E| \approx 170$ for $d = 3$. These are signatures of a phase transition in the thermodynamic limit. For $d = 1$ the Ising model does not have a phase transition.
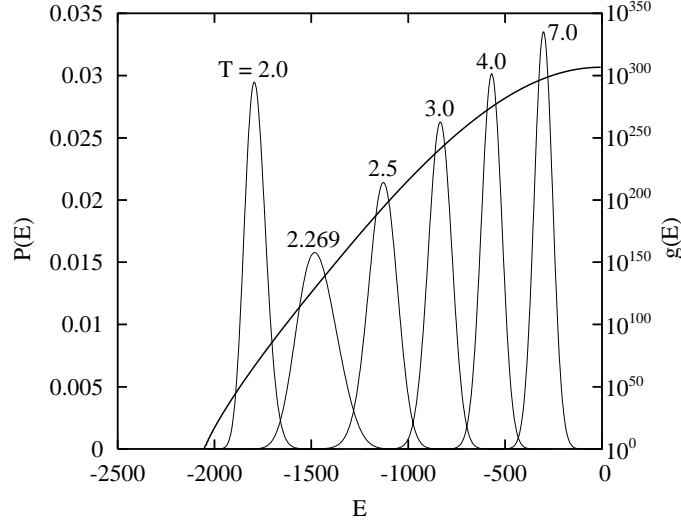
FIG. 2: The probability $P(E)$ of the energy E for the Ising model on a 32×32 square lattice for various temperatures. Superimposed on the same plot is the density of states.
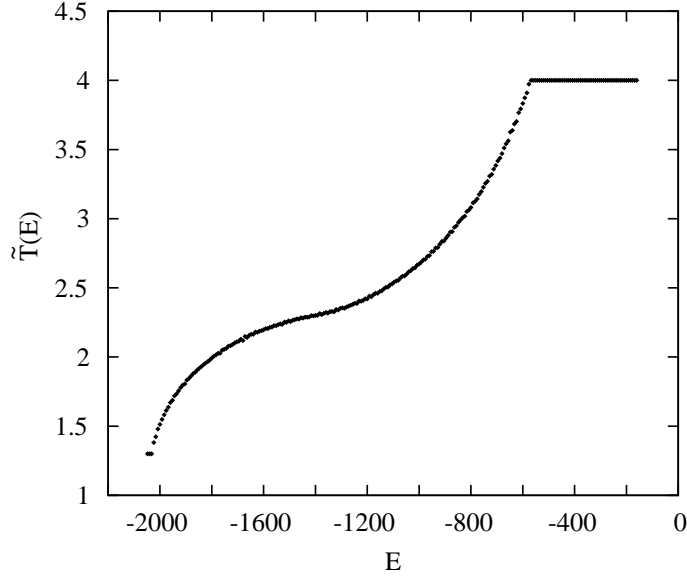

FIG. 3: The estimated energy-dependence of the temperature for the Ising model on a 32×32 square lattice as determined by the method of Ref. 14. The simulation was done with $T_{\max} = 4.5$ and $T_{\min} = 1.3$ using $\Delta E = 8$ and converged to $f = 1 + 10^{-12}$ after 4×10^6 Monte Carlo iterations per spin with the initial modification factor $f_0 = 1.00005$. The data was generated by Jaegil Kim.
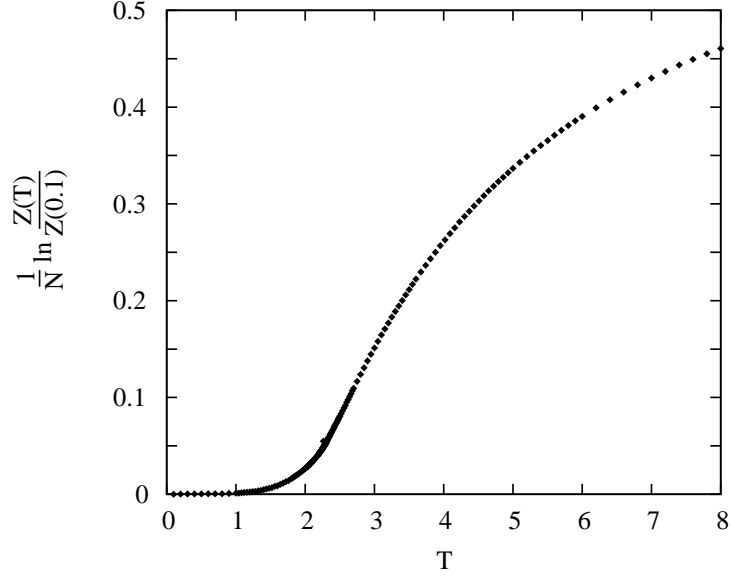
FIG. 4: Simulation results for $[\ln Z(T)/Z(0.1)]/N$ for a 32×32 Ising lattice. $T = 0.1$ is the lowest temperature simulated. The algorithm used and the temperatures simulated are the same as in Ref. 16. The final value of $\ln f \approx 3.16 \times 10^{-5}$. The number of MC steps at each value of f was $100N/\ln f$.